Claims

1. A method for detecting risky types of data structures of a computer program code with a neural network, said neural network comprising at least two neurons, and the neurons being related to each other by a topological arrangement involving a neighborhood definition, each of the neurons comprises a vector for representing elements of an input data space, at least one neuron having an associated label indicating the type of the neuron, and the data structures being detected comprising at least two data elements, **characterized** in that the method comprises,

extracting information of at least two data elements from at least one data structure,

forming at least two input vectors from said extracted information of the data elements, the vectors being compatible with the vectors of the neurons,

comparing said input vectors with said vectors of the neurons, and

detecting the type of said at least one data structure by using an associated label obtained on the basis of said comparison.

2. A method according to claim 1, **characterized** in that in the method said data vectors of the neurons have been formed by applying a self-organizing learning process, wherein learning data vectors have been allowed to change the vectors of the neurons by using a neighborhood mapping.

3. A method according to claim 1, **characterized** in that said neurons have been labeled on the basis of a labeling data item, wherein said labeling data item is examined to be at least one of the following; safe type data item, risky type data item or fail type data item.

4. A method according to claim 1, **characterized** in that in the method the input vector is compared to the vector of the neuron by using at least one of the following methods: the Euclidean distance, the Hamming distance, the Taxicab drivers distance, L1 norm, or dot product.

5. A method according to claim 4, **characterized** in that in the method the type of the data structure is detected by selecting the label of a neuron whose vector has the closest metric.

6. A method according to claim 4, **characterized** in that in the method the type of the data structure is detected by selecting the label of the closest neighbor of the neuron whose vector has the closest metric.

7. A method according to claim 4, **characterized** in that in the method the type of the data structure is detected by selecting the label of the closest labeled neuron on the map next to the neuron whose vector has the closest metric.

8. An electronic device (400) for detecting risky types of data structures of a computer program code with a neural network, said neural network comprising at least two neurons, and the neurons being related to each other by a topological arrangement involving a neighborhood definition, each of the neurons each comprises a vector for representing elements of an input data space, at least one neuron having an associated label indicating the type of the neuron, and the data structures being detected comprising at least two data elements, **characterized** in that the device comprises,

extracting means (401, 402, 406) for extracting information of at least two data elements from at least one data structure,

formation means (401, 402, 406) for forming at least two input vectors from said extracted information of the data elements, the vectors being compatible with the vectors of the neurons,

comparison means (401, 402, 406) for comparing said input vectors with said vectors of the neurons, and

detecting means (401, 402, 406) for detecting the type of said data structure by using an associated label obtained on the basis of said comparison.

9. A device according to claim 8, **characterized** in that the device further comprising second forming means (401, 402, 406) for forming said data vectors of the neurons by applying a self-organizing learning process, wherein learning data vectors have been allowed to change the vectors of the neurons by using a neighborhood mapping.

10. A device according to claim 9, **characterized** in that the second forming means (401, 402, 406) are further arranged to label the neurons on the basis of a labeling data item, wherein said labeling data item is examined to be at least one of the following; safe type data item, risky type data item or fail type data item.

11. A device according to claim 8, **characterized** in that in the comparison means (401, 402, 406) are arranged to compare the input vector to the vector of the neuron by using at least one of the following methods: the Euclidean distance the Hamming distance, the Taxicab drivers distance, L1 norm, or dot product.

12. A device according to claim 8 or 11, **characterized** in that the detecting means (401, 402, 406) are arranged to detect the type of the data structure by selecting the label of a neuron whose vector has the closest metric.

13. A device according to claim 8 or 11, **characterized** in that the detecting means (401, 402, 406) are arranged to detect the type of the data structure by selecting the label of the closest neighbor of the neuron whose vector has the closest metric.

14. A device according to claim 8 or 11, **characterized** in that the detecting means (401, 402, 406) are arranged to detect the type of the data structure by selecting the label of the closest labeled neuron on the map next to the neuron whose vector has the closest metric.

15. A computer program product for an electronic device for detecting risky types of data structures of a computer program code with a neural network, said neural

network comprising at least two neurons, and the neurons being related to each other by a topological arrangement involving a neighborhood definition, each of the neurons comprises a vector for representing elements of an input data space, at least one neuron having an associated label indicating the type of the neuron, and the data structures being detected comprising at least two data elements, **characterized** in that the computer program product comprises,

computer program code for causing the electronic device to extract information of at least two data elements from at least one data structure,

computer program code for causing the electronic device to form at least two input vectors from said extracted information of the data elements, the vectors being compatible with the vectors of the neurons,

computer program code for causing the electronic device to compare said input vectors with said vectors of the neurons, and

computer program code for causing the electronic device to detect the type of said data structure by using an associated label obtained on the basis of said comparison.